

▶ BootCamp Backend Java

Desafio Testing

// Prática Em Grupo

Objetivo

O objetivo deste desafio é aplicar os conteúdos dados até o momento durante o Programa de aceleração MeLi (Git, Java e Spring), com ênfase principal nas validações e tipos de teste que podem ser usados a partir de uma proposta, um especificação de requisitos e documentação anexa.

Diretrizes para atividade

O desafio proposto abaixo consiste em 2 partes:

A. Implemente validações e testes para um cenário determinado: A partir de um cenário conhecido, deverá se estabelecer os diferentes processos de validação de dados e processos de testes unitários necessários.

B. Bônus: Caso todos os requisitos estabelecidos no **ponto A** sejam cumpridos e ainda haja tempo disponível, esta atividade, que apresenta um nível de complexidade superior, pode ser realizada.

Dinâmica de trabalho

Te pedimos que, em primeiro lugar, responda este [Quiz inicial](#) para sabermos um pouco mais como você avalia o seu próprio conhecimento em relação ao conteúdo de testing.

A cada dia, duplas de trabalho serão organizadas para que, em algum momento do dia, seja dado um feedback ao colega sobre o projeto que está sendo realizado. O objetivo é trocar consultas, contribuições e valorizar o trabalho realizado pelo colega. Por sua vez, terão um **formulário** como rubrica, para registrar o feedback de cada dia.

Durante o desafio os professores estarão presentes através do meet ou Slack para eventuais dúvidas.

Tenha em conta que:

- O prazo de entrega e encerramento é: **segunda-feira, 4 de março, às 9h** e deve ser feito através do GitHub.
 - Todo o desafio deve ser desenvolvido em um **único projeto**.
 - Para entregar a atividade, o projeto concluído deve ser carregado para o repositório na pasta "**desafio_quality**".
- É **imprescindível** que o projeto seja carregado **COMPLETO**, ou seja, com todos os arquivos necessários para poder executá-lo.

A. Cenário

A empresa "**Seu Imóvel**" precisa ser capaz de calcular a metragem e o custo dos diferentes imóveis que possui em sua carteira de clientes.

Para isso, solicita de cada imóvel: um nome do imóvel, um bairro (nome do bairro e valor do metro quadrado no bairro) e a quantidade de cômodos que cada imóvel tem.

Ao mesmo tempo, cada cômodo contém um nome, uma largura e um comprimento.

Para isso, é necessário a criação de uma API Rest que permita:

- **US-0001:** Calcular a área total de uma propriedade.
- **US-0002:** Indicar o preço dessa mesma propriedade com base na área total. Lembre-se que os preços por metro quadrado são determinados em função do

bairro.

- **US-0003:** Determinar qual é o maior cômodo da propriedade.
- **US-0004:** Determinar a área de cada cômodo.

"**Seu Imóvel**" tem padrões de qualidade muito elevados no que diz respeito aos produtos de software que utiliza, uma vez que as transações que realiza diariamente são por quantias muito altas de dinheiro. É por isso que um consultor de informática que trabalha com eles estabeleceu uma série de validações que considera necessárias para levar em consideração na hora de incorporar dados, bem como diferentes testes de unidade que garantem os cálculos corretos.

Como documentação de apoio, em anexo o seguinte documento técnico funcional: [Requisitos Funcionais](#).

B. Bônus

Seguindo o princípio de que "**Seu Imóvel**" possui padrões de qualidade altos, o consultor de informática sugeriu a possibilidade de ter diferentes testes de integração, além de testes de unidade.

O consultor conhece os tempos limitados disponíveis para realizar o desenvolvimento solicitado, por isso sugere realizar esta implementação apenas no caso de cumprir os prazos para que a data de entrega estimada seja cumprida.